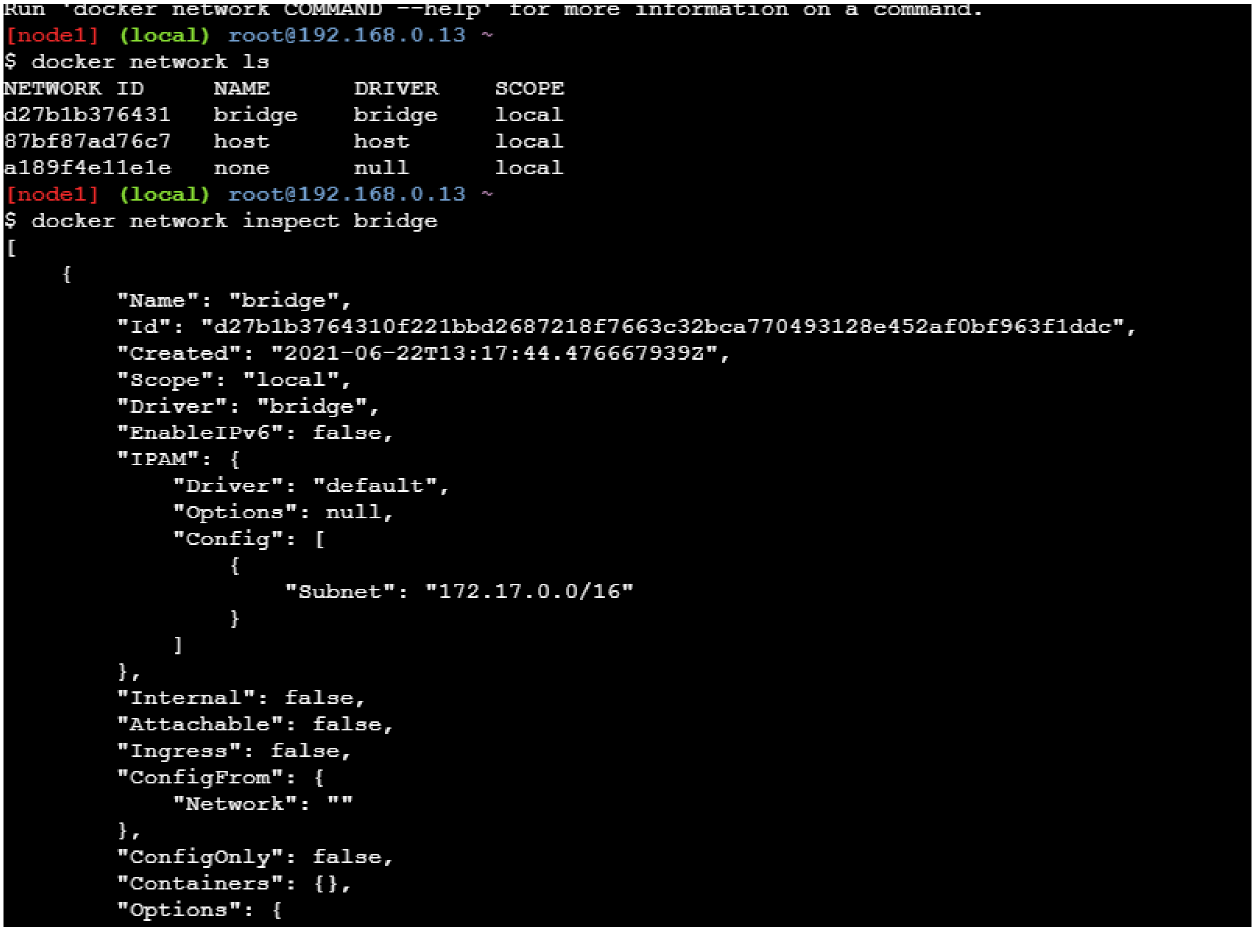
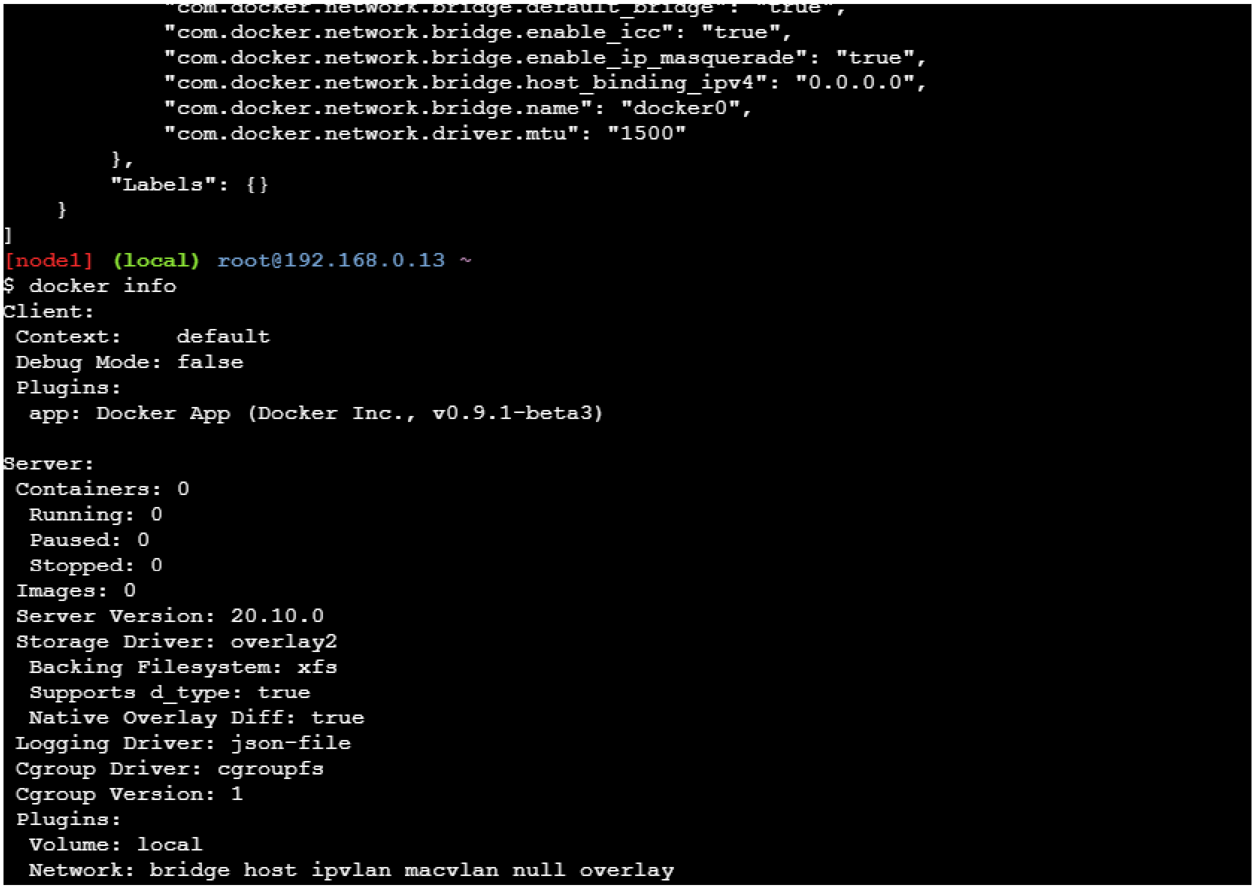
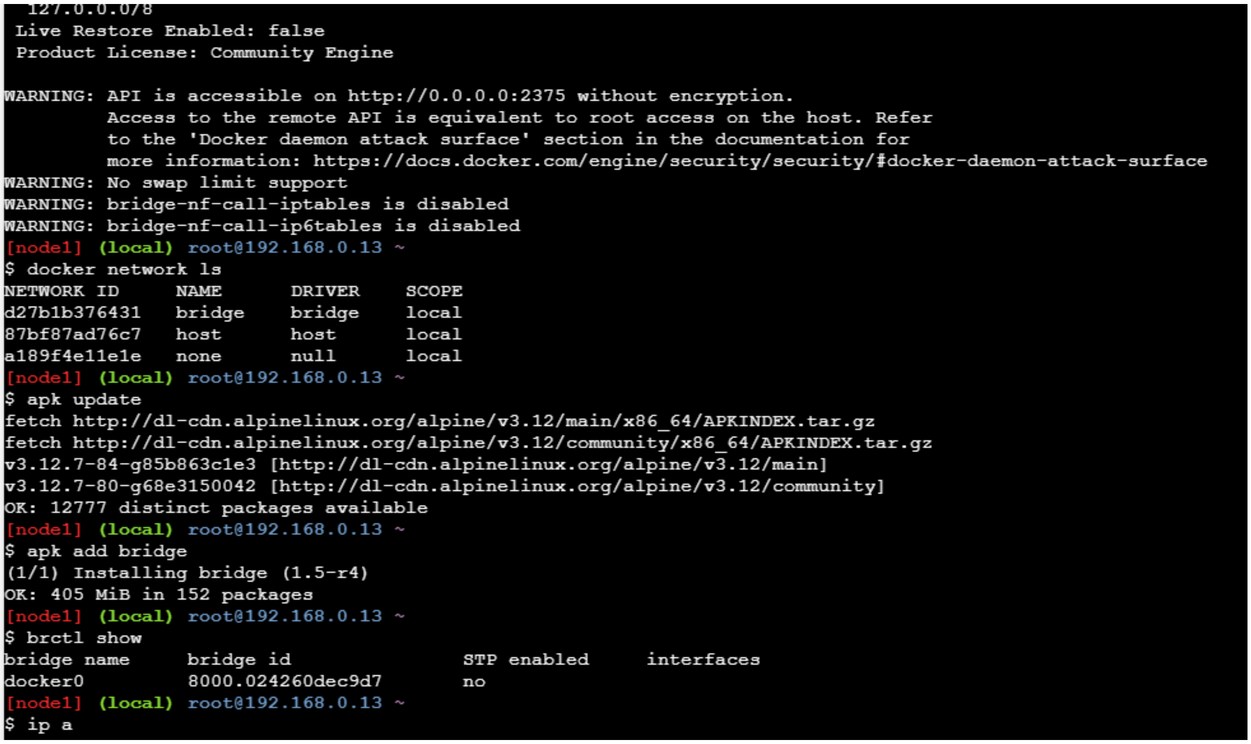
Note: I was told by sir darvy to perform this activity, Docker Networking Hands-on Lab (play-with-docker.com), for exercise 6 because of constraints with browser-based docker. The website operates using two small terminals. I used the big terminal for the first steps, and used the two small terminals for the swarming and service creation activities. Note that since the two terminals are small and there is a lot of data shown especially when inspecting networks, I decided to only show a portion of them so it would be easier to check.



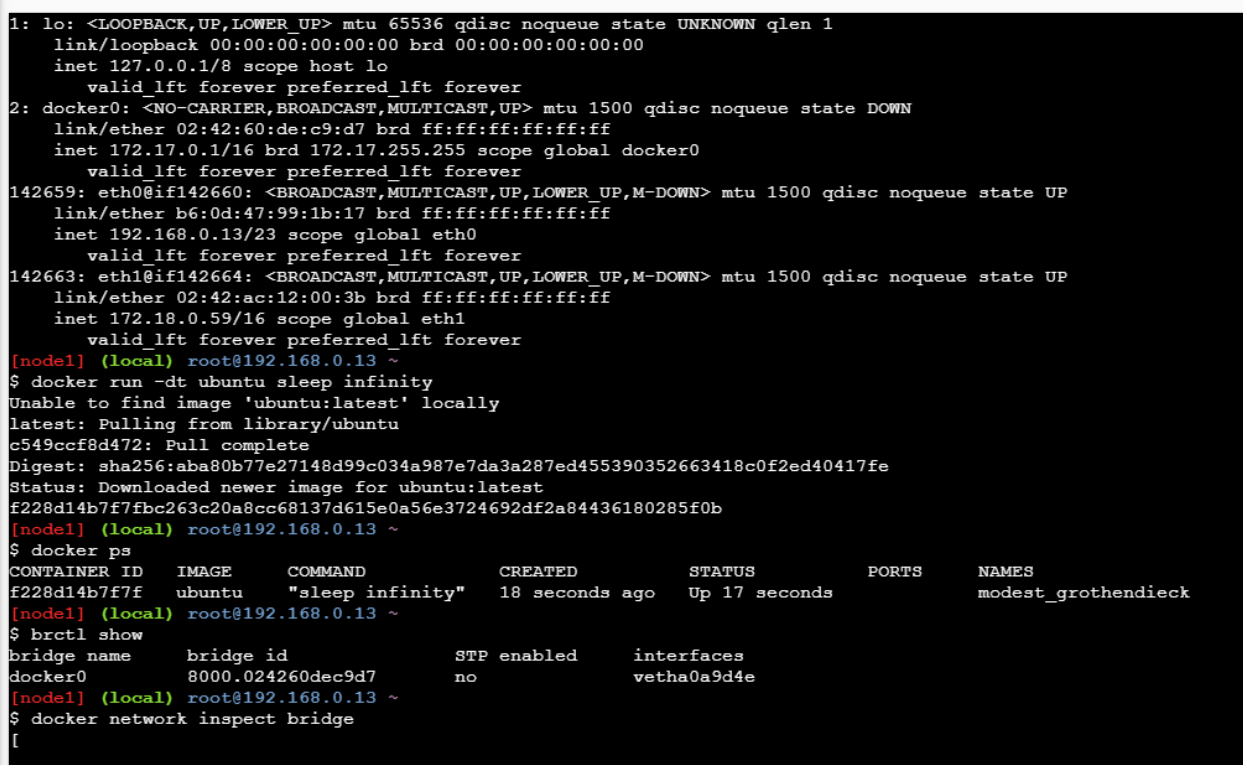
```
Run 'docker network COMMAND --help' for more information on a command.
[node1] (local) root@192.168.0.13 ~
$ docker network ls
NETWORK ID     NAME      DRIVER    SCOPE
d27b1b376431   bridge    bridge    local
87bf87ad76c7   host      host      local
a189f4e11e1e   none      null      local
[node1] (local) root@192.168.0.13 ~
$ docker network inspect bridge
[
    {
        "Name": "bridge",
        "Id": "d27b1b3764310f221bbd2687218f7663c32bca770493128e452af0bf963f1ddc",
        "Created": "2021-06-22T13:17:44.476667939Z",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": null,
            "Config": [
                {
                    "Subnet": "172.17.0.0/16"
                }
            ]
        },
        "Internal": false,
        "Attachable": false,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
        "Containers": {},
        "Options": {
```



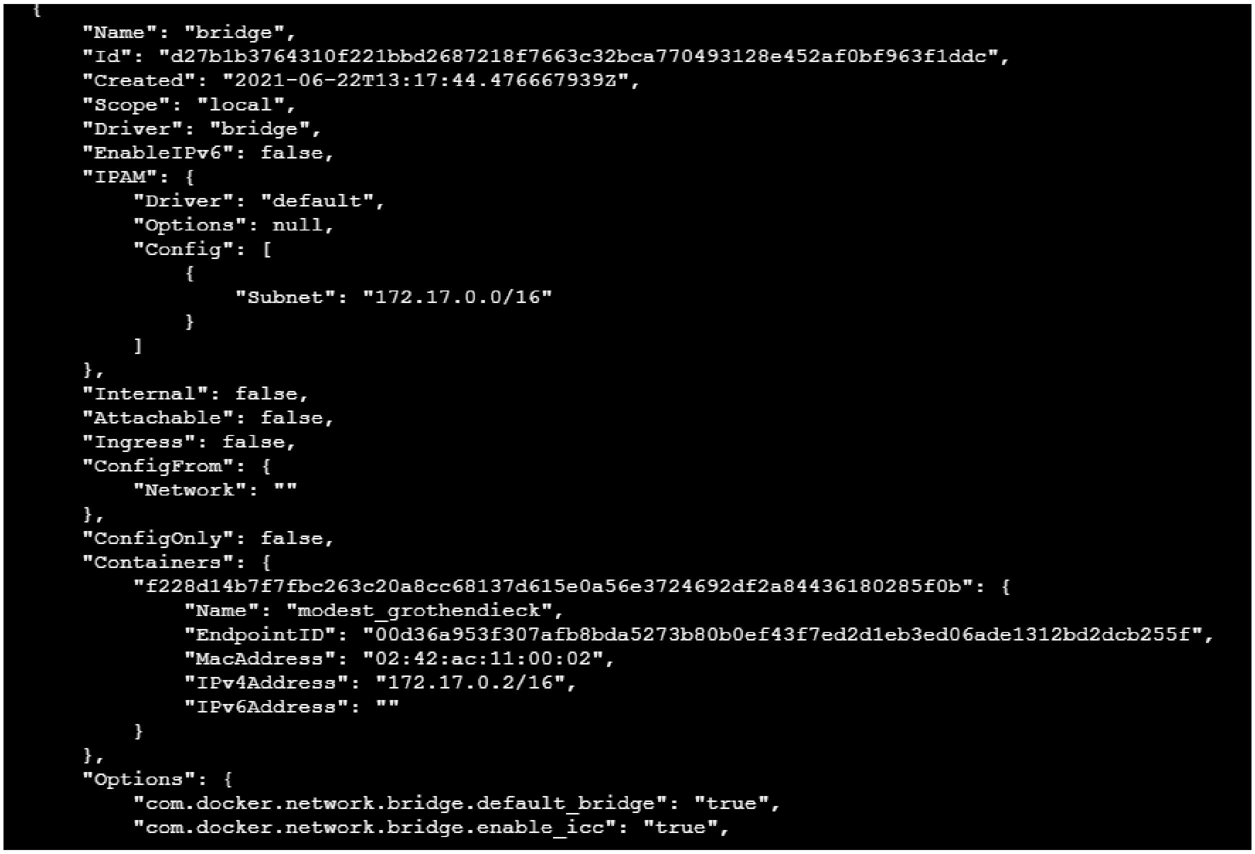
```
            "com.docker.network.bridge.default_bridge": "true",
            "com.docker.network.bridge.enable_icc": "true",
            "com.docker.network.bridge.enable_ip_masquerade": "true",
            "com.docker.network.bridge.host_binding_ipv4": "0.0.0.0",
            "com.docker.network.bridge.name": "docker0",
            "com.docker.network.driver.mtu": "1500"
        },
        "Labels": {}
    }
]
[node1] (local) root@192.168.0.13 ~
$ docker info
Client:
 Context:    default
 Debug Mode: false
 Plugins:
  app: Docker App (Docker Inc., v0.9.1-beta3)

Server:
 Containers: 0
  Running: 0
  Paused: 0
  Stopped: 0
 Images: 0
 Server Version: 20.10.0
 Storage Driver: overlay2
  Backing Filesystem: xfs
  Supports d_type: true
  Native Overlay Diff: true
 Logging Driver: json-file
 Cgroup Driver: cgroupfs
 Cgroup Version: 1
 Plugins:
  Volume: local
  Network: bridge host ipvlan macvlan null overlay
```



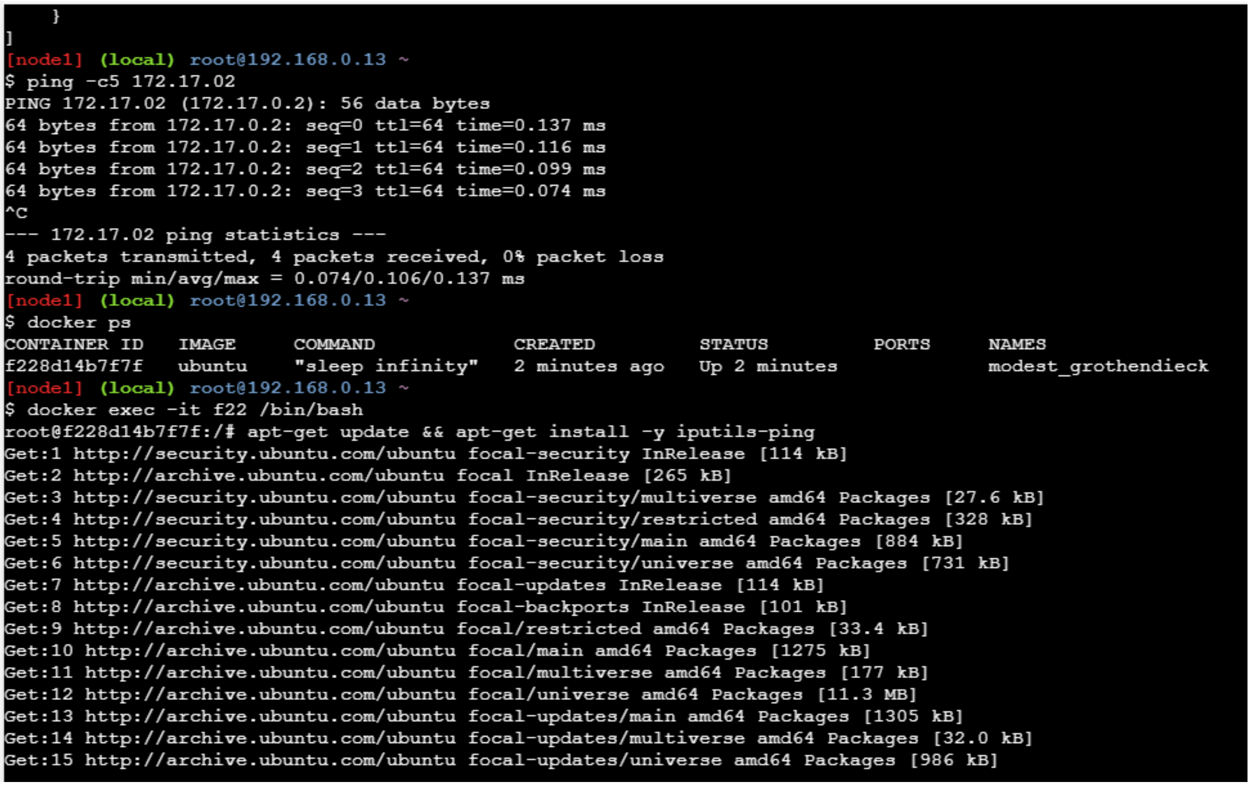
```
  127.0.0.0/8
 Live Restore Enabled: false
 Product License: Community Engine

WARNING: API is accessible on http://0.0.0.0:2375 without encryption.
         Access to the remote API is equivalent to root access on the host. Refer
         to the 'Docker daemon attack surface' section in the documentation for
         more information: https://docs.docker.com/engine/security/security/#docker-daemon-attack-surface
WARNING: No swap limit support
WARNING: bridge-nf-call-iptables is disabled
WARNING: bridge-nf-call-ip6tables is disabled
[node1] (local) root@192.168.0.13 ~
$ docker network ls
NETWORK ID     NAME      DRIVER    SCOPE
d27b1b376431   bridge    bridge    local
87bf87ad76c7   host      host      local
a189f4e11e1e   none      null      local
[node1] (local) root@192.168.0.13 ~
$ apk update
fetch http://dl-cdn.alpinelinux.org/alpine/v3.12/main/x86_64/APKINDEX.tar.gz
fetch http://dl-cdn.alpinelinux.org/alpine/v3.12/community/x86_64/APKINDEX.tar.gz
v3.12.7-84-g85b863c1e3 [http://dl-cdn.alpinelinux.org/alpine/v3.12/main]
v3.12.7-80-g68e3150042 [http://dl-cdn.alpinelinux.org/alpine/v3.12/community]
OK: 12777 distinct packages available
[node1] (local) root@192.168.0.13 ~
$ apk add bridge
(1/1) Installing bridge (1.5-r4)
OK: 405 MiB in 152 packages
[node1] (local) root@192.168.0.13 ~
$ brctl show
bridge name     bridge id               STP enabled     interfaces
docker0         8000.024260dec9d7       no
[node1] (local) root@192.168.0.13 ~
$ ip a
```



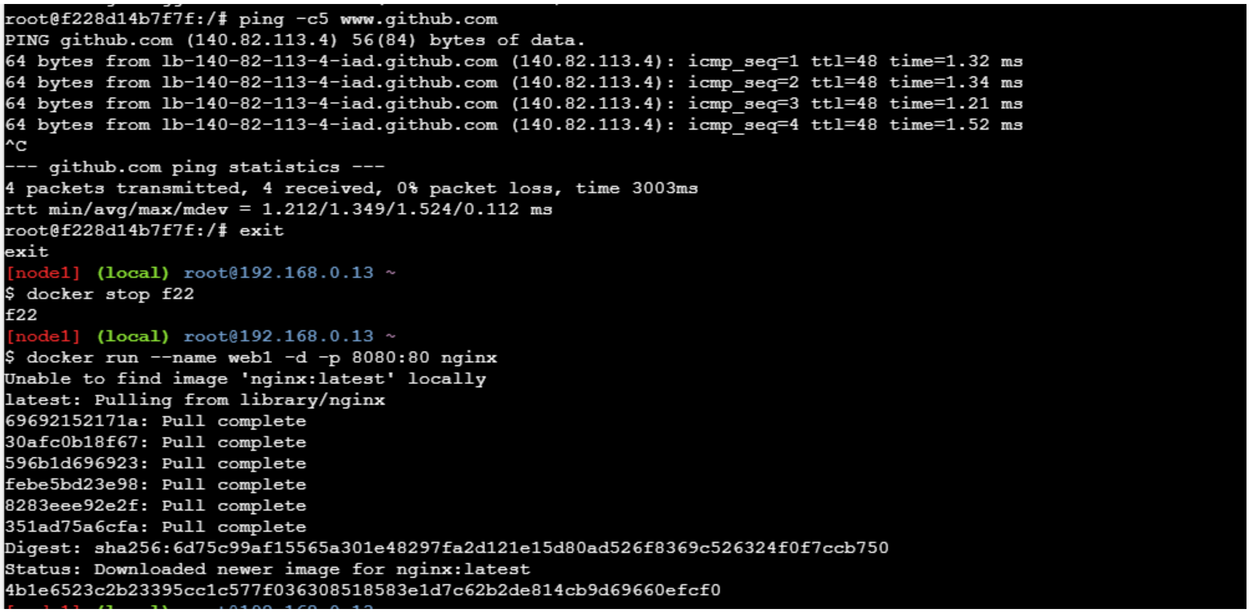
```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
2: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN
    link/ether 02:42:60:de:c9:d7 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
       valid_lft forever preferred_lft forever
142659: eth0@if142660: <BROADCAST,MULTICAST,UP,LOWER_UP,M-DOWN> mtu 1500 qdisc noqueue state UP
    link/ether b6:0d:47:99:1b:17 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.13/23 scope global eth0
       valid_lft forever preferred_lft forever
142663: eth1@if142664: <BROADCAST,MULTICAST,UP,LOWER_UP,M-DOWN> mtu 1500 qdisc noqueue state UP
    link/ether 02:42:ac:12:00:3b brd ff:ff:ff:ff:ff:ff
    inet 172.18.0.59/16 scope global eth1
       valid_lft forever preferred_lft forever
[node1] (local) root@192.168.0.13 ~
$ docker run -dt ubuntu sleep infinity
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
c549ccf8d472: Pull complete
Digest: sha256:aba80b77e27148d99c034a987e7da3a287ed455390352663418c0f2ed40417fe
Status: Downloaded newer image for ubuntu:latest
f228d14b7f7fbc263c20a8cc68137d615e0a56e3724692df2a84436180285f0b
[node1] (local) root@192.168.0.13 ~
$ docker ps
CONTAINER ID   IMAGE     COMMAND             CREATED          STATUS          PORTS     NAMES
f228d14b7f7f   ubuntu    "sleep infinity"    18 seconds ago   Up 17 seconds             modest_grothendieck
[node1] (local) root@192.168.0.13 ~
$ brctl show
bridge name     bridge id               STP enabled     interfaces
docker0         8000.024260dec9d7       no              vetha0a9d4e
[node1] (local) root@192.168.0.13 ~
$ docker network inspect bridge
[
```



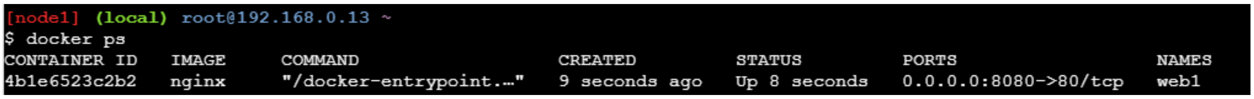
```
    {
        "Name": "bridge",
        "Id": "d27b1b3764310f221bbd2687218f7663c32bca770493128e452af0bf963f1ddc",
        "Created": "2021-06-22T13:17:44.476667939Z",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": null,
            "Config": [
                {
                    "Subnet": "172.17.0.0/16"
                }
            ]
        },
        "Internal": false,
        "Attachable": false,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
        "Containers": {
            "f228d14b7f7fbc263c20a8cc68137d615e0a56e3724692df2a84436180285f0b": {
                "Name": "modest_grothendieck",
                "EndpointID": "00d36a953f307afb8bda5273b80b0ef43f7ed2d1eb3ed06ade1312bd2dcb255f",
                "MacAddress": "02:42:ac:11:00:02",
                "IPv4Address": "172.17.0.2/16",
                "IPv6Address": ""
            }
        },
        "Options": {
            "com.docker.network.bridge.default_bridge": "true",
            "com.docker.network.bridge.enable_icc": "true",
```



```
    }
]
[node1] (local) root@192.168.0.13 ~
$ ping -c5 172.17.02
PING 172.17.02 (172.17.0.2): 56 data bytes
64 bytes from 172.17.0.2: seq=0 ttl=64 time=0.137 ms
64 bytes from 172.17.0.2: seq=1 ttl=64 time=0.116 ms
64 bytes from 172.17.0.2: seq=2 ttl=64 time=0.099 ms
64 bytes from 172.17.0.2: seq=3 ttl=64 time=0.074 ms
^C
--- 172.17.02 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 0.074/0.106/0.137 ms
[node1] (local) root@192.168.0.13 ~
$ docker ps
CONTAINER ID   IMAGE     COMMAND              CREATED         STATUS         PORTS     NAMES
f228d14b7f7f   ubuntu    "sleep infinity"     2 minutes ago   Up 2 minutes             modest_grothendieck
[node1] (local) root@192.168.0.13 ~
$ docker exec -it f22 /bin/bash
root@f228d14b7f7f:/# apt-get update && apt-get install -y iputils-ping
Get:1 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:2 http://archive.ubuntu.com/ubuntu focal InRelease [265 kB]
Get:3 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 Packages [27.6 kB]
Get:4 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 Packages [328 kB]
Get:5 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [884 kB]
Get:6 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [731 kB]
Get:7 http://archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:8 http://archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
Get:9 http://archive.ubuntu.com/ubuntu focal/restricted amd64 Packages [33.4 kB]
Get:10 http://archive.ubuntu.com/ubuntu focal/main amd64 Packages [1275 kB]
Get:11 http://archive.ubuntu.com/ubuntu focal/multiverse amd64 Packages [177 kB]
Get:12 http://archive.ubuntu.com/ubuntu focal/universe amd64 Packages [11.3 MB]
Get:13 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [1305 kB]
Get:14 http://archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 Packages [32.0 kB]
Get:15 http://archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [986 kB]
```

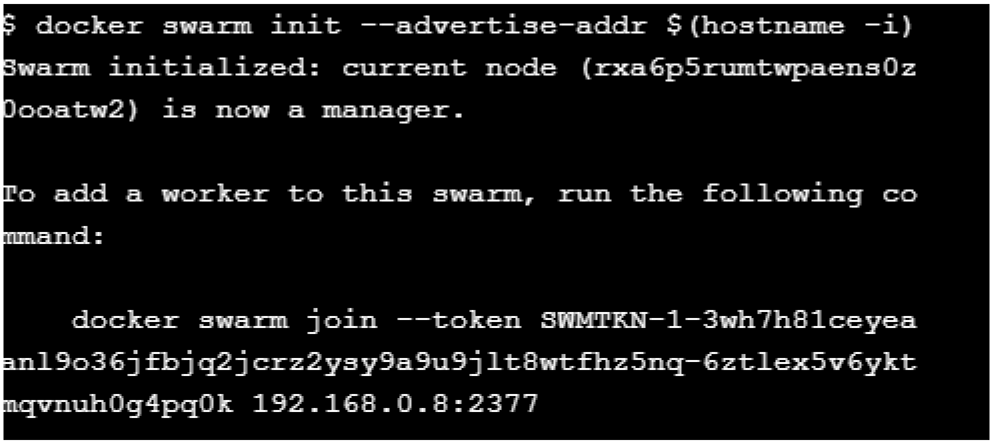


```
root@f228d14b7f7f:/# ping -c5 www.github.com
PING github.com (140.82.113.4) 56(84) bytes of data.
64 bytes from lb-140-82-113-4-iad.github.com (140.82.113.4): icmp_seq=1 ttl=48 time=1.32 ms
64 bytes from lb-140-82-113-4-iad.github.com (140.82.113.4): icmp_seq=2 ttl=48 time=1.34 ms
64 bytes from lb-140-82-113-4-iad.github.com (140.82.113.4): icmp_seq=3 ttl=48 time=1.21 ms
64 bytes from lb-140-82-113-4-iad.github.com (140.82.113.4): icmp_seq=4 ttl=48 time=1.52 ms
^C
--- github.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3003ms
rtt min/avg/max/mdev = 1.212/1.349/1.524/0.112 ms
root@f228d14b7f7f:/# exit
exit
[node1] (local) root@192.168.0.13 ~
$ docker stop f22
f22
[node1] (local) root@192.168.0.13 ~
$ docker run --name web1 -d -p 8080:80 nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
69692152171a: Pull complete
30afc0b18f67: Pull complete
596b1d696923: Pull complete
febe5bd23e98: Pull complete
8283eee92e2f: Pull complete
351ad75a6cfa: Pull complete
Digest: sha256:6d75c99af15565a301e48297fa2d121e15d80ad526f8369c526324f0f7ccb750
Status: Downloaded newer image for nginx:latest
4b1e6523c2b23395cc1c577f036308518583e1d7c62b2de814cb9d69660efcf0
```

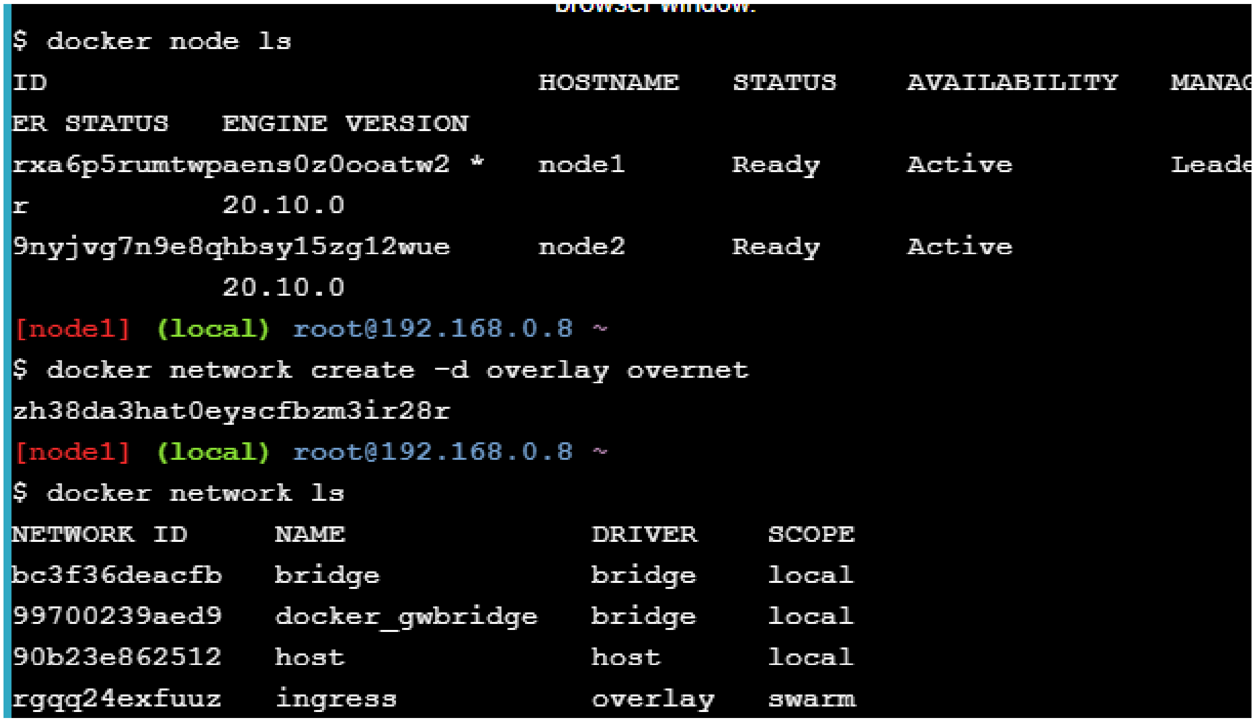


```
[node1] (local) root@192.168.0.13 ~
$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS         PORTS                  NAMES
4b1e6523c2b2   nginx     "/docker-entrypoint.…"   9 seconds ago   Up 8 seconds   0.0.0.0:8080->80/tcp   web1
```

1st teminal:

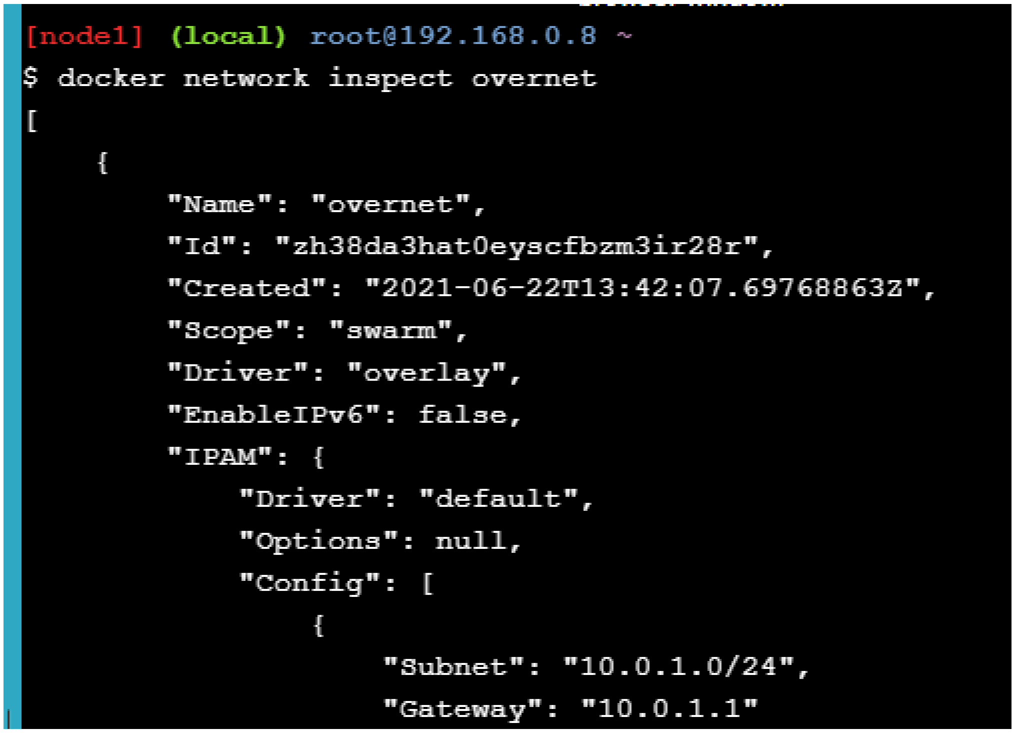


```
$ docker swarm init --advertise-addr $(hostname -i)
Swarm initialized: current node (rxa6p5rumtwpaens0z
0ooatw2) is now a manager.

To add a worker to this swarm, run the following co
mmand:

    docker swarm join --token SWMTKN-1-3wh7h81ceyea
anl9o36jfbjq2jcrz2ysy9a9u9jlt8wtfhz5nq-6ztlex5v6ykt
mqvnuh0g4pq0k 192.168.0.8:2377
```

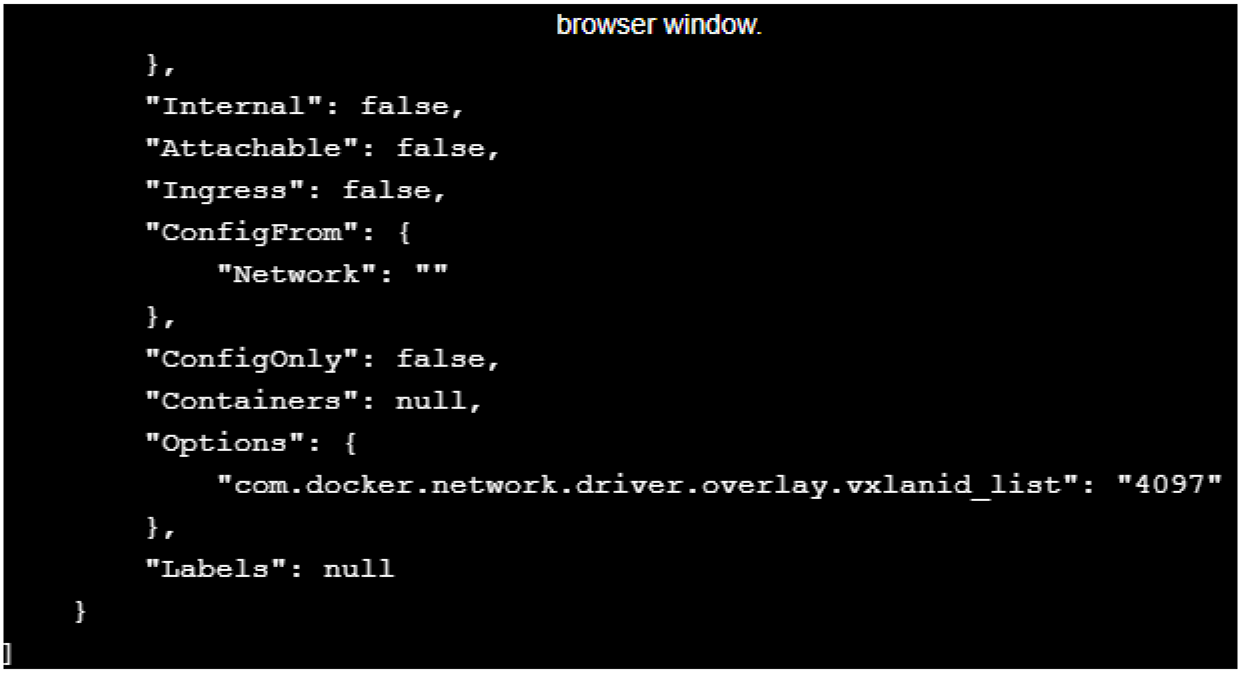


browser window.

```
$ docker node ls
ID                            HOSTNAME   STATUS    AVAILABILITY   MANAG
ER STATUS   ENGINE VERSION
rxa6p5rumtwpaens0z0ooatw2 *   node1      Ready     Active         Leade
r           20.10.0
9nyjvg7n9e8qhbsy15zg12wue     node2      Ready     Active
            20.10.0
[node1] (local) root@192.168.0.8 ~
$ docker network create -d overlay overnet
zh38da3hat0eyscfbzm3ir28r
[node1] (local) root@192.168.0.8 ~
$ docker network ls
NETWORK ID     NAME              DRIVER    SCOPE
bc3f36deacfb   bridge            bridge    local
99700239aed9   docker_gwbridge   bridge    local
90b23e862512   host              host      local
rgqq24exfuuz   ingress           overlay   swarm
```

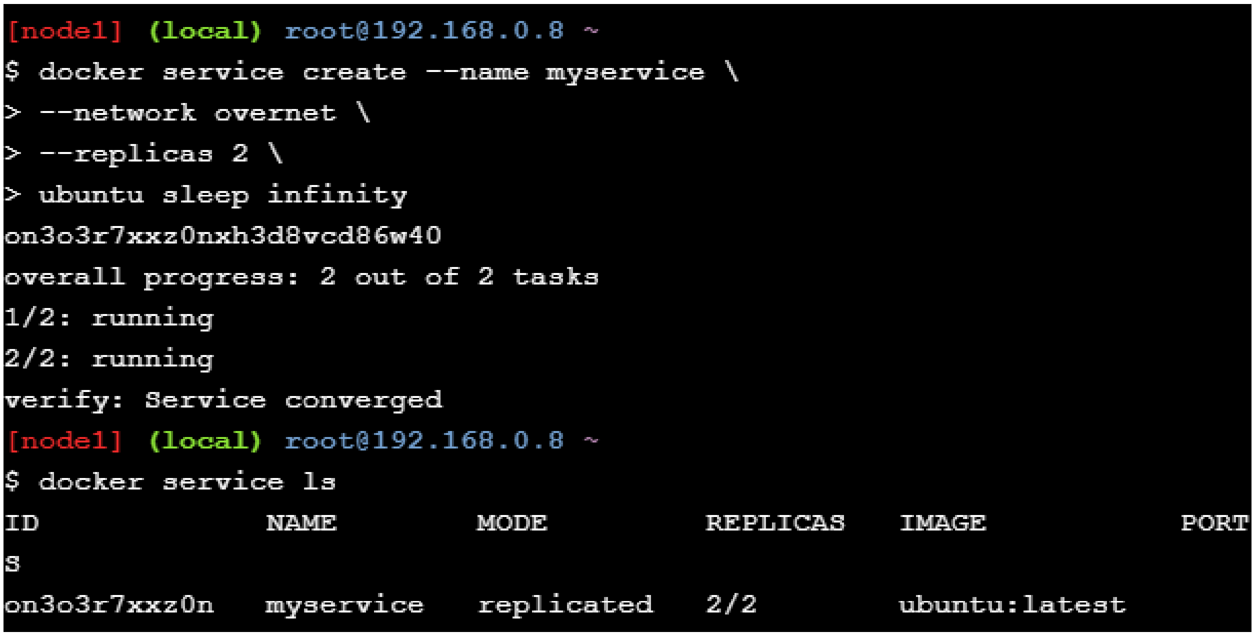


```
[node1] (local) root@192.168.0.8 ~
$ docker network inspect overnet
[
    {
        "Name": "overnet",
        "Id": "zh38da3hat0eyscfbzm3ir28r",
        "Created": "2021-06-22T13:42:07.697688632Z",
        "Scope": "swarm",
        "Driver": "overlay",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": null,
            "Config": [
                {
                    "Subnet": "10.0.1.0/24",
                    "Gateway": "10.0.1.1"
```

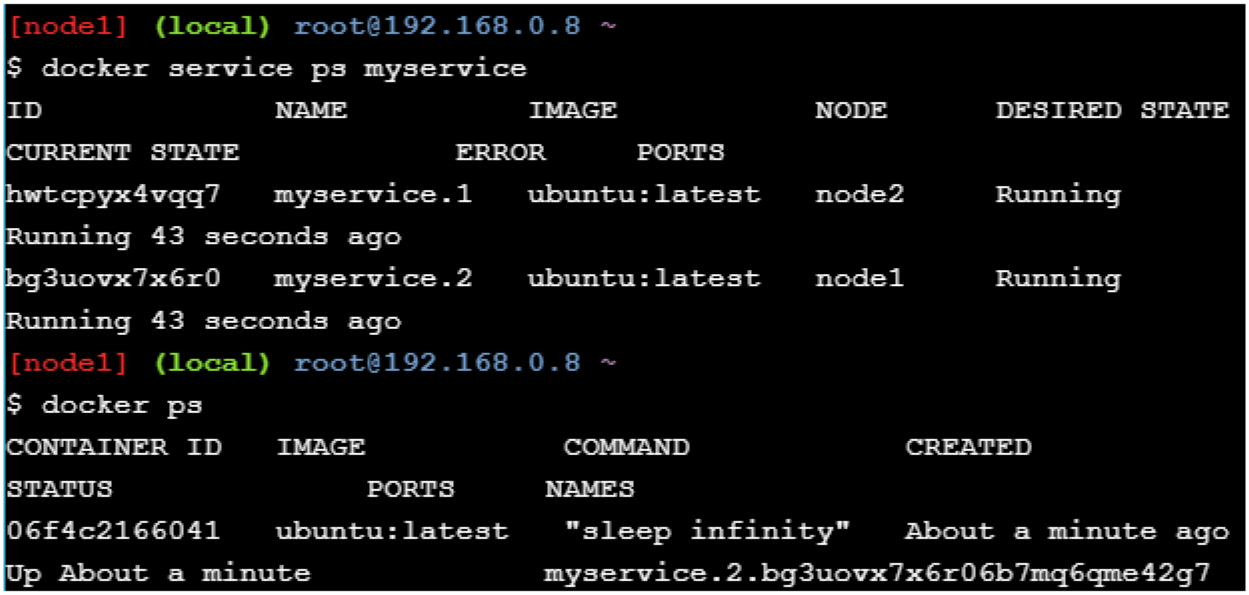


browser window.

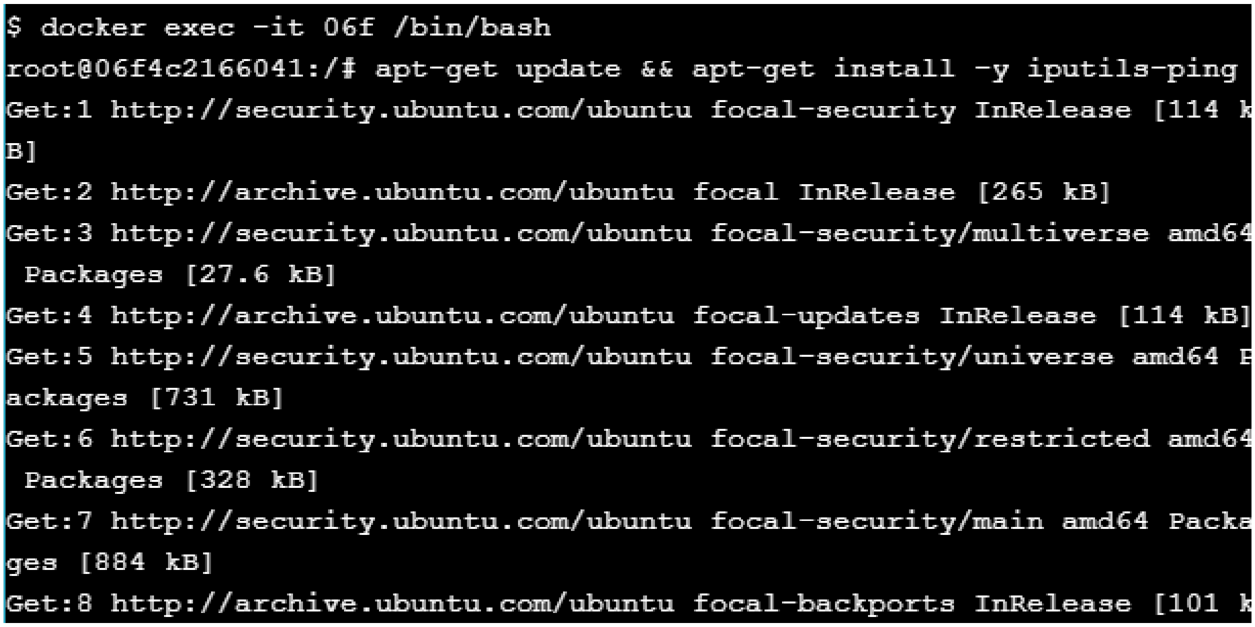
```
        },
        "Internal": false,
        "Attachable": false,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
        "Containers": null,
        "Options": {
            "com.docker.network.driver.overlay.vxlanid_list": "4097"
        },
        "Labels": null
    }
]
```



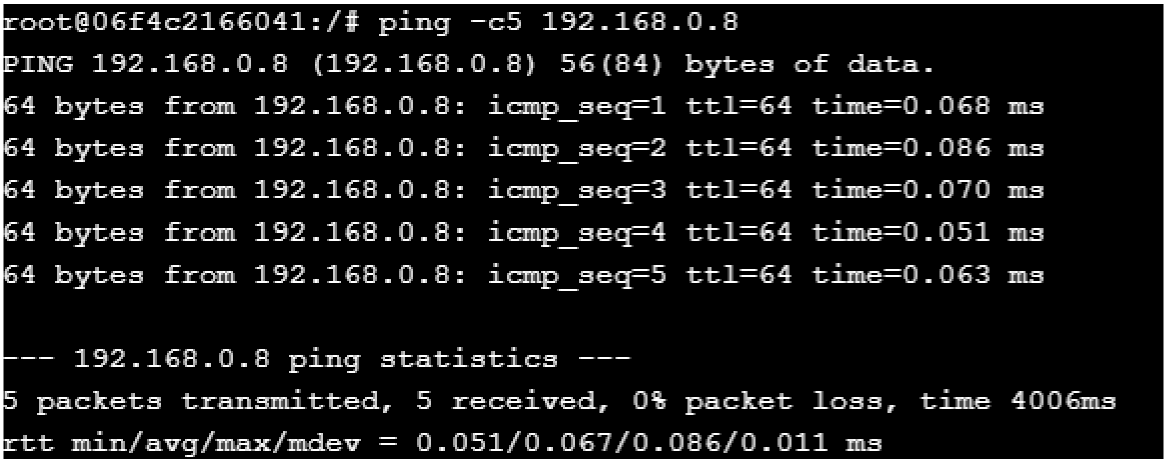
```
[node1] (local) root@192.168.0.8 ~
$ docker service create --name myservice \
> --network overnet \
> --replicas 2 \
> ubuntu sleep infinity
on3o3r7xxz0nxh3d8vcd86w40
overall progress: 2 out of 2 tasks
1/2: running
2/2: running
verify: Service converged
[node1] (local) root@192.168.0.8 ~
$ docker service ls
ID             NAME        MODE         REPLICAS   IMAGE           PORT
S
on3o3r7xxz0n   myservice   replicated   2/2        ubuntu:latest
```



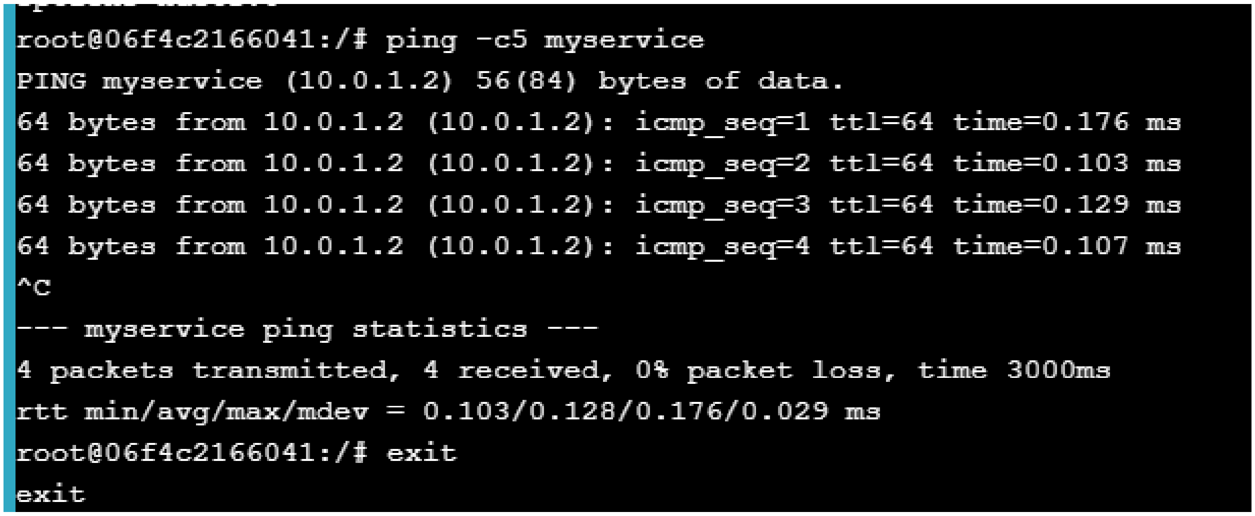
```
[node1] (local) root@192.168.0.8 ~
$ docker service ps myservice
ID             NAME          IMAGE           NODE      DESIRED STATE
CURRENT STATE            ERROR     PORTS
hwtcpyx4vqq7   myservice.1   ubuntu:latest   node2     Running
Running 43 seconds ago
bg3uovx7x6r0   myservice.2   ubuntu:latest   node1     Running
Running 43 seconds ago
[node1] (local) root@192.168.0.8 ~
$ docker ps
CONTAINER ID   IMAGE            COMMAND            CREATED
STATUS              PORTS     NAMES
06f4c2166041   ubuntu:latest    "sleep infinity"   About a minute ago
Up About a minute             myservice.2.bg3uovx7x6r06b7mq6qme42g7
```



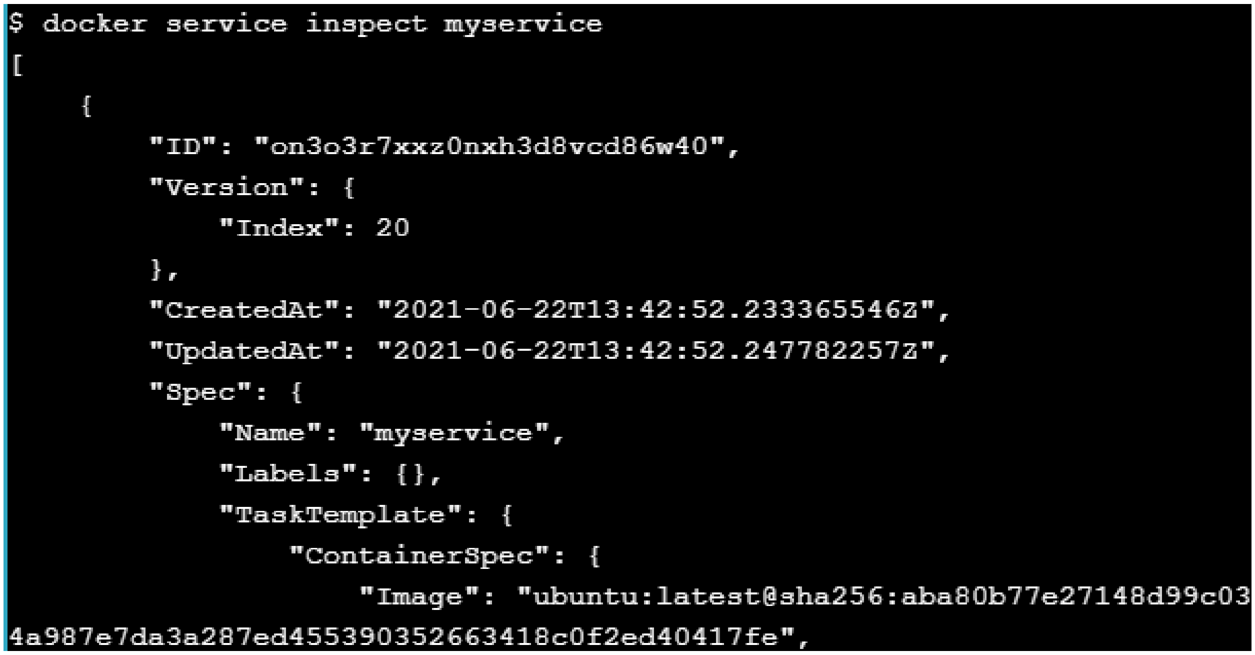
```
$ docker exec -it 06f /bin/bash
root@06f4c2166041:/# apt-get update && apt-get install -y iputils-ping
Get:1 http://security.ubuntu.com/ubuntu focal-security InRelease [114 k
B]
Get:2 http://archive.ubuntu.com/ubuntu focal InRelease [265 kB]
Get:3 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64
 Packages [27.6 kB]
Get:4 http://archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:5 http://security.ubuntu.com/ubuntu focal-security/universe amd64 P
ackages [731 kB]
Get:6 http://security.ubuntu.com/ubuntu focal-security/restricted amd64
 Packages [328 kB]
Get:7 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packa
ges [884 kB]
Get:8 http://archive.ubuntu.com/ubuntu focal-backports InRelease [101 k
```



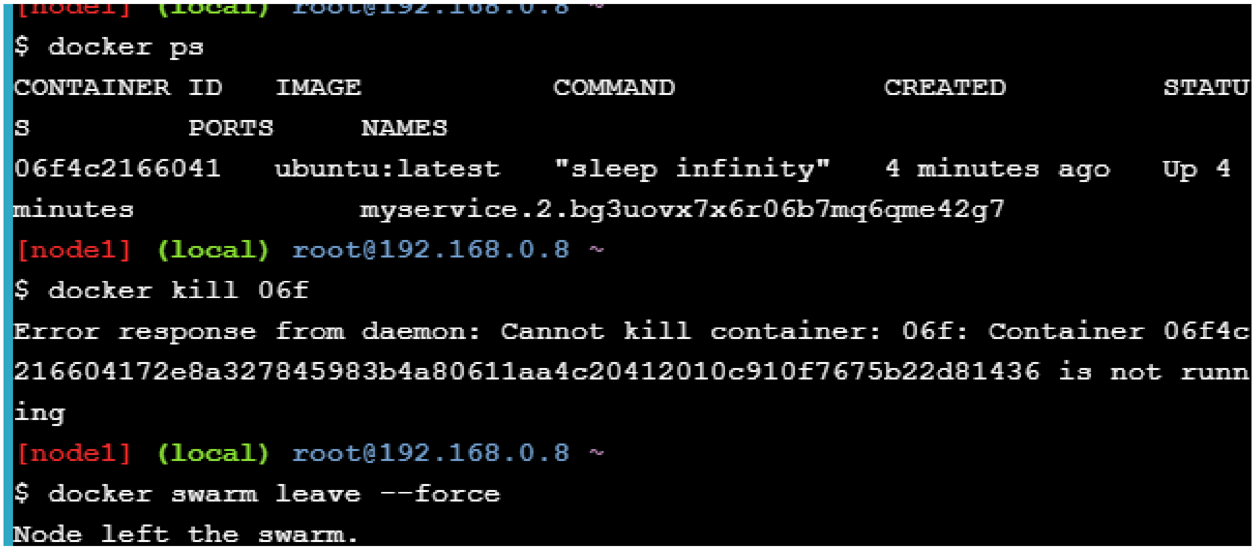
```
root@06f4c2166041:/# ping -c5 192.168.0.8
PING 192.168.0.8 (192.168.0.8) 56(84) bytes of data.
64 bytes from 192.168.0.8: icmp_seq=1 ttl=64 time=0.068 ms
64 bytes from 192.168.0.8: icmp_seq=2 ttl=64 time=0.086 ms
64 bytes from 192.168.0.8: icmp_seq=3 ttl=64 time=0.070 ms
64 bytes from 192.168.0.8: icmp_seq=4 ttl=64 time=0.051 ms
64 bytes from 192.168.0.8: icmp_seq=5 ttl=64 time=0.063 ms

--- 192.168.0.8 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 0.051/0.067/0.086/0.011 ms
```



```
root@06f4c2166041:/# ping -c5 myservice
PING myservice (10.0.1.2) 56(84) bytes of data.
64 bytes from 10.0.1.2 (10.0.1.2): icmp_seq=1 ttl=64 time=0.176 ms
64 bytes from 10.0.1.2 (10.0.1.2): icmp_seq=2 ttl=64 time=0.103 ms
64 bytes from 10.0.1.2 (10.0.1.2): icmp_seq=3 ttl=64 time=0.129 ms
64 bytes from 10.0.1.2 (10.0.1.2): icmp_seq=4 ttl=64 time=0.107 ms
^C
--- myservice ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3000ms
rtt min/avg/max/mdev = 0.103/0.128/0.176/0.029 ms
root@06f4c2166041:/# exit
exit
```

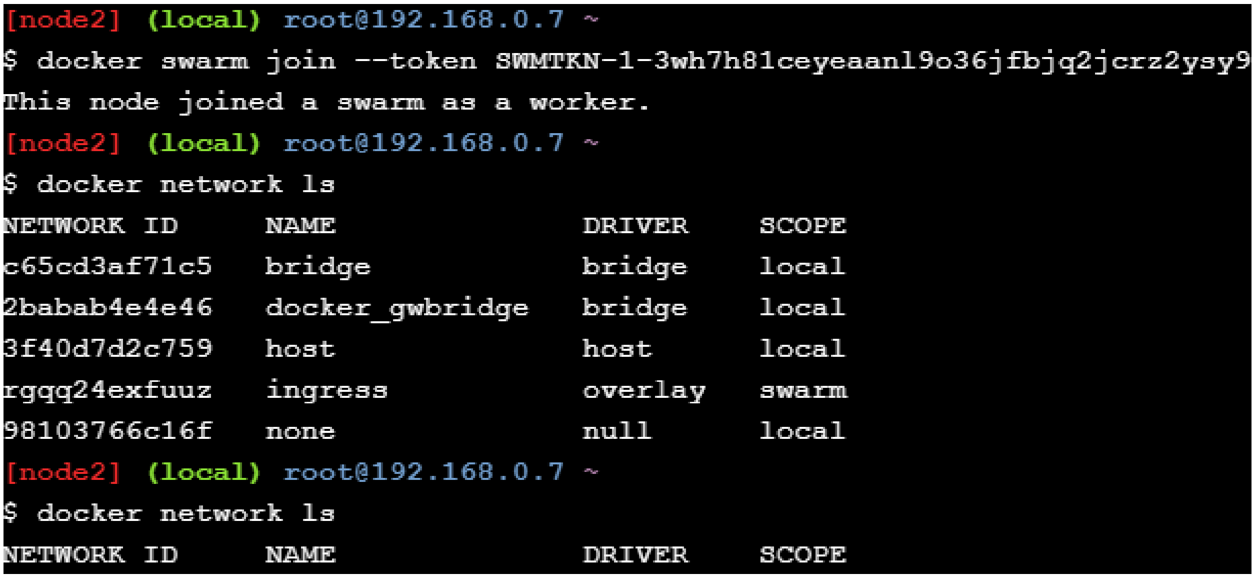


```
$ docker service inspect myservice
[
    {
        "ID": "on3o3r7xxz0nxh3d8vcd86w40",
        "Version": {
            "Index": 20
        },
        "CreatedAt": "2021-06-22T13:42:52.233365546Z",
        "UpdatedAt": "2021-06-22T13:42:52.247782257Z",
        "Spec": {
            "Name": "myservice",
            "Labels": {},
            "TaskTemplate": {
                "ContainerSpec": {
                    "Image": "ubuntu:latest@sha256:aba80b77e27148d99c03
4a987e7da3a287ed455390352663418c0f2ed40417fe",
```

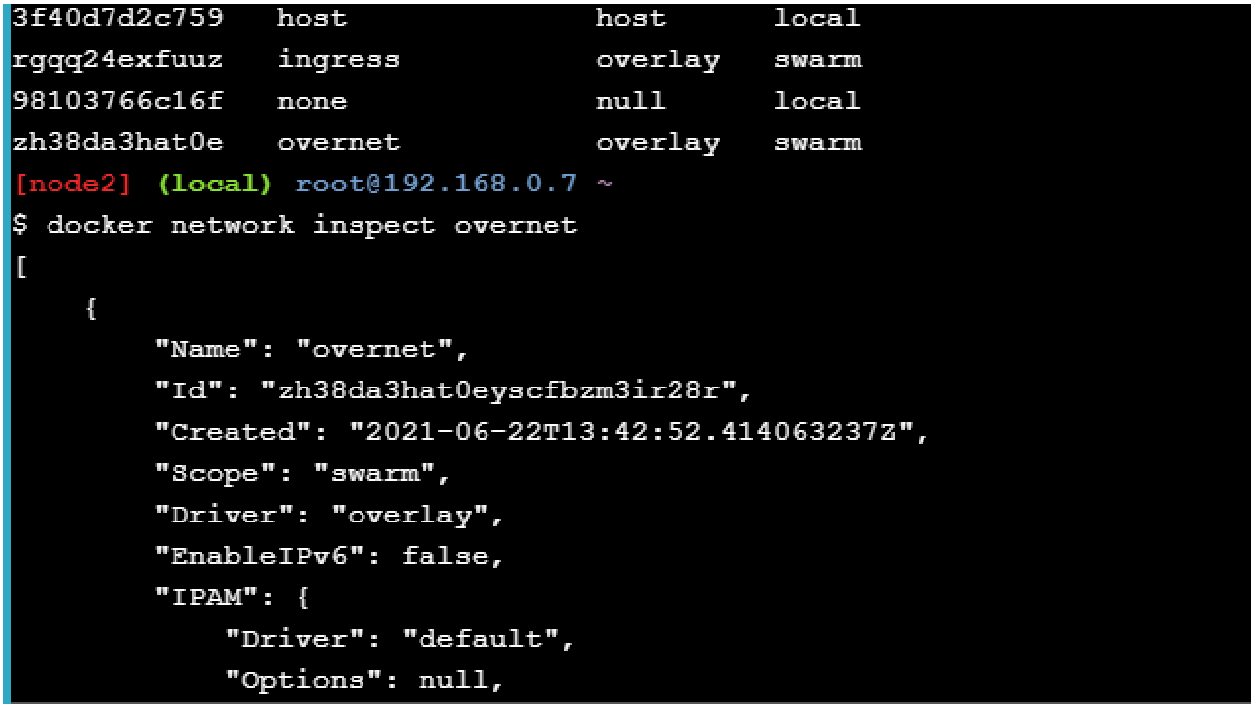


```
[node1] (local) root@192.168.0.8 ~
$ docker ps
CONTAINER ID   IMAGE           COMMAND             CREATED          STATU
S          PORTS     NAMES
06f4c2166041   ubuntu:latest   "sleep infinity"   4 minutes ago    Up 4
minutes              myservice.2.bg3uovx7x6r06b7mq6qme42g7
[node1] (local) root@192.168.0.8 ~
$ docker kill 06f
Error response from daemon: Cannot kill container: 06f: Container 06f4c
216604172e8a327845983b4a80611aa4c20412010c910f7675b22d81436 is not runn
ing
[node1] (local) root@192.168.0.8 ~
$ docker swarm leave --force
Node left the swarm.
```

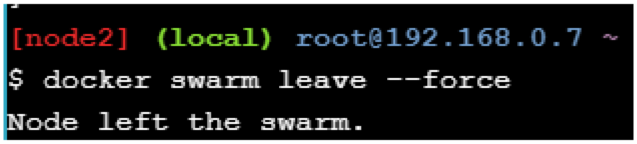
2nd terminal:



```
[node2] (local) root@192.168.0.7 ~
$ docker swarm join --token SWMTKN-1-3wh7h81ceyeaanl9o36jfbjq2jcrz2ysy9
This node joined a swarm as a worker.
[node2] (local) root@192.168.0.7 ~
$ docker network ls
NETWORK ID     NAME              DRIVER    SCOPE
c65cd3af71c5   bridge            bridge    local
2babab4e4e46   docker_gwbridge   bridge    local
3f40d7d2c759   host              host      local
rgqq24exfuuz   ingress           overlay   swarm
98103766c16f   none              null      local
[node2] (local) root@192.168.0.7 ~
$ docker network ls
NETWORK ID     NAME              DRIVER    SCOPE
```



```
3f40d7d2c759   host              host      local
rgqq24exfuuz   ingress           overlay   swarm
98103766c16f   none              null      local
zh38da3hat0e   overnet           overlay   swarm
[node2] (local) root@192.168.0.7 ~
$ docker network inspect overnet
[
    {
        "Name": "overnet",
        "Id": "zh38da3hat0eyscfbzm3ir28r",
        "Created": "2021-06-22T13:42:52.414063237Z",
        "Scope": "swarm",
        "Driver": "overlay",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": null,
```



```
[node2] (local) root@192.168.0.7 ~
$ docker swarm leave --force
Node left the swarm.
```